

REMARKS/ARGUMENTS

These remarks are submitted in response to the Office Action of November 21, 2006 (hereinafter Office Action). As this response is timely filed within the 3-month shortened statutory period, no fee is believed due. Nonetheless, the Examiner is expressly authorized to charge any deficiencies or credit any overpayment to Deposit Account No. 50-0951.

In paragraph 3, page 2 of the Office Action, each of Claims 1-3, 5-9, 11-15, 17 & 18 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Application Publication No. 2002/0076008A1 to Neary (hereinafter Neary) in view of U.S. Patent Application Publication No. 2002/0077819A1 to Girardo (hereinafter Girardo).

Applicants have amended independent Claims 1, 7, and 13 to further emphasize certain aspects of the invention. Applicants also have amended dependent Claims 3, 9, and 15 to maintain consistency among all the remaining claims. As discussed herein, the claim amendments are fully supported throughout the Specification. No new matter has been introduced through the claim amendments presented.

Applicants' Invention

At this juncture, it may be useful to reiterate certain aspects of Applicants' invention. One embodiment of the invention, typified by amended Claim 1, is a method of verifying software program operations during execution of a voice response system comprising establishing a voice link between a test system and the voice response system, executing one or more operational software programs in the voice response system to determine a voice prompt to play over the established voice link, performing speech recognition at the test system to speech recognize the voice prompt played over the established link and convert the voice prompt to text, gathering execution information at

the voice response system associated with the one or more executing operational software programs, wherein the execution information identifies computer programs and modules used by the voice response system during the step of executing one or more operational software programs, and sending the execution information to the test system over the voice link, wherein the execution information pertains to the execution of the one or more operational software programs on the voice response system.

The Claims Define Over The Prior Art

On Page 3 of the Office Action, claims 1-3 were rejected as being unpatentable over Neary in view of Girardo. Briefly, Applicants have amended the claims to further distinguish aspects of the invention. One of the differences between Applicants' invention and Neary's invention is the execution information. In the rejection of claim 1, it was specifically stated that Neary discloses execution information ("one", "two", etc.; see, e.g., Specification, Paragraph [0018]) for the test system to process and route the call to the correct voice path by sending multiple audible speech to the test system over the voice link. Notably, the execution information described by Neary is related to either the input commands (e.g., the test system audibly presenting "one", "two", etc., or presenting DTMF tones to the VRS; See FIG. 2), or the outcome of the input commands (e.g., coded signals or DTMF tones that identify the utterance presented by the VRS in response to an input command; See, e.g., Specification, FIG. 2). Neary is concerned with execution information that is external to the VRS. It should be noted that Neary is not concerned with the processes that occur internal to the VRS during the execution of the input commands at the VRS or with identifying the processes occurring internal to the VRS.

In contrast, one aspect of Applicants' invention is the identifying of internal execution processes in the VRS. In particular, the execution information utilized with Applicants' invention identifies computer programs and modules used by the VRS during the step of executing one or more operational software programs (See, e.g., Specification,

Paragraph [0021])). The execution information allows the test system to identify programs in the VRS that were called during the step of executing. This allows the test system to determine which programs were called at the VRS in response to the input commands presented to the VRS. Moreover, this allows test system engineers to identify the programs used by the VRS and the program order in the VRS. As an example, a program path address can be specified as a uniform resource locator URL (See, e.g., Specification Paragraph [0021])). The execution information can identify methods and functions used by the VRS during the execution of one or more input commands. It should be noted that Neary does not teach gathering execution information that identifies computer programs and modules used during execution at the VRS. Neary only teaches evaluating an outcome response (See paragraph [0018]; see also Fig. 2). The outcome response provided by Neary does not contain any information related to the processes or programs running internal to the VRS.

The execution information is one of the differences that clearly distinguishes Applicants' invention from Neary. Again, Neary is directed to evaluating call-flow verification which consists of transmitting input commands to a VRS and then evaluating the output of the VRS. The output of the VRS is compared to outputs known to be correct, which allows the test system to determine if the call flow is correct. For example, Neary includes an encoder 30 in the VRS and a decoder 34 in the test system (ACG) to verify call-flow. The ACG is pre-programmed for call flow verification by storage of data representing of content the VRS is intended to provide (Paragraph [0018])). The encoder merely compares the predetermined known correct output with the VRS generated output and sends a coded signal that identifies whether the output was correct or incorrect. In such capacity, Neary is limited to attempting to determine whether the output is correct or incorrect. Neary does not teach or suggest anything about the programs or modules within the VRS used to generate the output. Neary does not even contemplate internal operations of the VRS, let alone identifying programs or modules executing internal to

the VRS that produce an output. In contrast, Applicants' invention enables the monitoring of internal processes performed by the VRS during the execution of input commands. In particular, Applicants' invention is able to identify and report computer programs and modules to the test system.

On Page 5 of the Office Action, with respect to the rejection of claim 3, it was stated that Neary verifies whether calls are routed to the correct voice path according to input response from the test system based on comparison analysis. Applicants respectfully submit that the term "path" should be evaluated in the context of Neary's written description (See, e.g., Abstract where it is stated that "verification can be applied for all possible paths an application might take (to check all possible go-right and error paths)"). In such regard, Neary is merely stating that a VRS has a hierarchy of paths which can be traversed for responding to an input command, which is common knowledge among those skilled in the art. The call verification flow proposed by Neary simply determines whether all voice response paths are operating correctly. Clearly, these paths are not related to computer program path addresses or method call paths used by the VRS, such as a uniform resource locator URI or a program directory structure (See Specification Paragraph [0021]). Again, the test system of Applicants' invention can identify computer programs and modules by path name address used during execution to identify the source of the execution.

In the rejection of claim 1, it was also stated that Neary fails to disclose converting the voice prompt to text, though Girardo discloses a testing and verification system that would obviate converting a voice prompt to text. In particular, it was stated that Neary teaches an IVR system to send audible speech to the test system in response to received execution information. In such regard, the Examiner has assumed that the execution information is associated with the input commands to the VRS. In the context of Neary's invention, this is correct, since the execution information is related to the input commands presented to the VRS. However, in the context of Applicants' invention, the

execution information is generated by the VRS after the input commands are received. (Note: in Neary's case, if the execution information is received by the VRS, then it cannot be the same execution information generated by the VRS, as in Applicants' invention).

As an example, the VRS of Applicants' invention may perform voice recognition on an input command. The Voice Recognition operation on the VRS may perform a series of method calls to evaluate a plurality of speech tokens, each identified as a separate file on a directory path. The VRS may also call a number of program functions. During execution, the VRS can identify the directory structure for the called speech tokens, the called programs, or any other processing information and save them to a temporary database during execution (See database 115, FIG. 1; see also paragraph [0021]). The execution information can then be included in the execution information sent from the VRS to the test system. The test system can then identify which methods and voice tokens were used during recognition by the path address name. Notably, this provides the test system with comprehensive information related to the program execution that occurs internal to the VRS. Again, this is one of the aspects of Applicants' invention which Neary fails to consider or contemplate. Neary does not contemplate sending execution information from the VRS to the test system that identifies computer programs and modules used by the voice response system during the step of executing one or more operational software programs.

Accordingly, Neary in view of Girardo does not teach or suggest any aspect of performing speech recognition related to gathering execution information related to computer programs and modules that are specified by path address name and method call names. Applicants, therefore, respectfully submit that the references, even when combined, fail to teach or suggest every feature recited in amended independent Claims 1, 7, and 13, and that each of the claims thus defines over the prior art. Applicants further respectfully submit that whereas each of the remaining dependent claims depends

from one of the amended claims while reciting additional features, the dependent claims likewise define over the prior art.

CONCLUSION

Applicants believe that this application is now in full condition for allowance, which action is respectfully requested. Applicants request that the Examiner call the undersigned if clarification is needed on any matter within this Amendment, or if the Examiner believes a telephone interview would expedite the prosecution of the subject application to completion.

Respectfully submitted,

Date: February 21, 2007



Gregory A. Nelson, Registration No. 30,577

Richard A. Hinson, Registration No. 47,652

Marc A. Boillot, Registration No. 56,164

AKERMAN SENTERFITT

Customer No. 40987

Post Office Box 3188

West Palm Beach, FL 33402-3188

Telephone: (561) 653-5000